

AD-A167 453

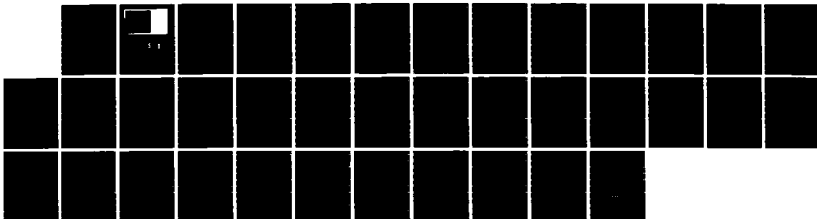
TREE-STRUCTURED CLASSIFICATION VIA GENERALIZED  
DISCRIMINANT ANALYSIS(U) WISCONSIN UNIV-MADISON  
MATHEMATICS RESEARCH CENTER W LOH ET AL. JAN 86  
ARC-TSR-2982 DRAG29-88-C-0041

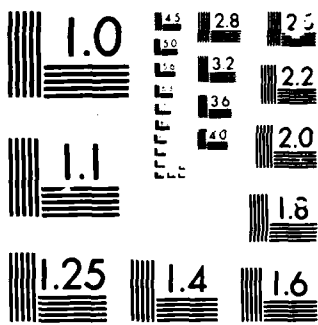
1/1

UNCLASSIFIED

F/G 9/2

NL





MICROCOPY

CHART

2

MRC Technical Summary Report #2902

TREE-STRUCTURED CLASSIFICATION VIA  
GENERALIZED DISCRIMINANT ANALYSIS

Wei-Yin Loh and N. Vanichsetakul

AD-A167 453

Mathematics Research Center  
University of Wisconsin—Madison  
610 Walnut Street  
Madison, Wisconsin 53705

January 1986

(Received January 17, 1986)

DTIC  
ELECTE  
MAY 23 1986  
S D

Approved for public release  
Distribution unlimited

Sponsored by

U. S. Army Research Office  
P. O. Box 12211  
Research Triangle Park  
North Carolina 27709

National Science Foundation  
Washington, DC 20550

86 5 20 151

DTIC FILE COPY

12

**UNIVERSITY OF WISCONSIN—MADISON  
MATHEMATICS RESEARCH CENTER**

**TREE-STRUCTURED CLASSIFICATION  
VIA GENERALIZED DISCRIMINANT ANALYSIS**

**Wei-Yin Loh and N. Vanichsetakul**

**Technical Summary Report # 2902**

January 1986

**Abstract**

Linear techniques are used recursively to construct classification rules which can be represented as  $k$ -nary decision trees. The method has been implemented in a computer program called FACT. It can handle ordered and unordered variables, unequal priors, variable misclassification costs, and missing observations. Besides the tree structure, it also yields an importance ranking of the variables and a cross-validation estimate of error. FACT is compared with CART (a procedure proposed recently by Breiman et al., which gives a binary tree) in a series of examples. The conclusion is that FACT and CART are usually comparable in terms of classification accuracy and interpretative capability, but FACT runs many times faster.

AMS (MOS) Subject Classification: 62H30, 68G10.

**KEY WORDS:** Classification trees; Cross-validation; Linear discriminant analysis; Misclassification costs; Missing values; Principal components; Recursive partitioning.

**Work Unit Number 4 — Statistics and Probability**

---

Department of Statistics, University of Wisconsin, Madison, WI 53706.

---

Loh's research was partially supported by National Science Foundation Grants MCS 8300140 and DMS 8502303, and partially sponsored by the United States Army under contract No. DAAG29-80-C-0041. Vanichsetakul's research was partially supported by funds from the Research Committee of the University of Wisconsin-Madison Graduate School. All the computations were done using the University of Wisconsin Department of Statistics Pyramid 90X superminicomputer. The first author is indebted to Professor L. Breiman for teaching him CART.

# TREE-STRUCTURED CLASSIFICATION VIA GENERALIZED DISCRIMINANT ANALYSIS

Wei-Yin Loh and N. Vanichsetakul

## 1. INTRODUCTION

### 1.1. The Problem

The problem considered is as follows: We have a  $k$ -vector of measurements  $\mathbf{x} = (x_1, \dots, x_k)'$  on an object which we know belongs to one of  $J$  classes, and wish to predict its class membership. It is assumed that we have at hand a "learning sample" of  $n$  other objects, where, in addition to their measurement vectors, we also know their actual class memberships. This problem is variously referred to as *discrimination* (Hand 1981), *identification* (Gordon 1981, p. 3), or *classification* (Breiman et al. 1984). We adopt the latter term here.

As might be expected, the scope of the problem is very broad. Some examples mentioned in Hand (1981, Section 1.1) are: (a) remote sensing of crops using high altitude photographs, (b) medical diagnosis based on health history and vital measurements, and (c) speech recognition via waveform data. Additional applications, discussed in Breiman et al., include (d) military ship identification from radar profiles, (e) analysis of chemical compounds via mass spectra, and (f) prediction of tomorrow's weather using current-day observations.

### 1.2. Some Solutions

In order to predict the class membership of a new object, a *classification rule* (or *classifier*) is needed. The classifier is usually constructed from information extracted from the learning sample, as well as any available information about prior probabilities of the classes and

Department of Statistics, University of Wisconsin, Madison, WI 53706.

Loh's research was partially supported by National Science Foundation Grants MCS 8300140 and DMS 8502303, and partially sponsored by the United States Army under contract No. DAAG29-80-C-0041. Vanichsetakul's research was partially supported by funds from the Research Committee of the University of Wisconsin-Madison Graduate School. All the computations were done using the University of Wisconsin Department of Statistics Pyramid 90X superminicomputer. The first author is indebted to Professor L. Breiman for teaching him CART.



Community Codes	
Dist	Avail and/or Special
A-1	

misclassification costs. If the true probability densities of the  $\mathbf{x}$ -vectors in the classes are known, the rule with the best error rate is the *Bayes* or *maximum likelihood rule*. Because these densities are hardly ever known in real problems, however, most of the available methods try to estimate them from the learning sample.

The earliest and best known method is *Fisher's linear discriminant analysis (LDA)*. If the  $\mathbf{x}$ -vectors are normally distributed and the covariance matrices are the same for all the  $J$  classes, this method is asymptotically Bayes. The LDA rule also has an intuitive interpretation. If we assume equal priors and unit misclassification costs, and the data has been first sphericized, LDA partitions the transformed  $\mathbf{x}$ -space into  $J$  disjoint portions, with each portion containing the estimated centroid for one class. The boundaries are such that every point in one portion is closer in Euclidean distance to the centroid contained in it, than to the other centroids. (See e.g. Gnanadesikan 1977, Chapter 4.) Because of this data-analytic property, LDA often performs satisfactorily even when the data are not normally distributed.

Other methods estimate the class probability densities nonparametrically, such as methods based on histogram and kernel density estimation, and nearest neighbor techniques. (An excellent survey of the literature prior to 1981 is given in Hand 1981.) While all these methods can be effective, they have been criticized on the following grounds (see Breiman et al. p. 17):

- (1) they are quite dependent on the metric  $\|\mathbf{x}\|$  used,
- (2) they do not handle categorical variables and missing observations naturally,
- (3) they are computationally expensive to use because the learning sample must be stored and then recalled every time a new object is to be classified, and
- (4) they act like "black boxes", yielding little useful information about the data structure.

Breiman et al. develop a novel method designed to avoid these criticisms. Their method, called *CART*, produces a classifier which can be represented as a *binary decision tree*. (Any partition of the variable-space can be represented in the form of a tree structure. The idea of

classification trees dates back to the *AID* and *THAID* programs developed at the University of Michigan by Morgan, Sonquist and Messenger in the early 1960s and 1970s; see e.g. Morgan and Messenger 1973 and Fielding 1977.) A new object is classified by dropping it down the tree. Its path down is determined from the answers to a series of questions, one at each node. Because the tree is the classifier, the learning sample may be discarded after the tree is constructed. Hence new objects can be classified quickly.

### 1.3. Potential Difficulties with CART

Despite CART's great flexibility, our experience with it reveals the following drawbacks:

- (1) CART chooses its splits at each node very meticulously, using exhaustive searches. When  $n$  or  $k$  is large, it can take a long time to construct the tree. The problem is worse with linear combination splits.
- (2) Compared to CART, LDA runs very fast. Further, as observed in Breiman et al. (Section 5.7), LDA tends to do quite well on many nonnormal data sets. Thus, if cost of classifier construction is important but the form of the classifier is not, LDA may be preferred.
- (3) In CART, a very large tree is initially constructed and then "pruned" upward, using cross-validation (CV), to arrive at the final tree. If the data is noisy enough (e.g. if a lot of observations are missing), CV may produce a very short tree. Now if, in addition,  $J$  is big, the tree may not have at least one terminal node for each class; i.e. some classes are unrepresented in the classifier.
- (4) Another disadvantage with this dual use of CV—for pruning and error rate estimation—is that the CART program cannot be made to run faster if a CV estimate of error is not desired. (For example, one may already be satisfied with CART for one learning sample, and wishes to quickly revise the tree with additional learning samples.)

- (5) Finally, because CART can typically only afford to use 10- or 25-fold CV to prune the tree, the resulting classifier is essentially a *randomized* rule. That is, if a different random number seed is used to divide the data for the purpose of CV, a different-sized tree may result (although the splits at each node remain the same).

#### 1.4. The Proposed Solution

Motivated by the above observations, we propose a different solution—called *FACT* (for “Fast and Automatic Classification Tree”). Our original goal was to combine the best features of LDA and CART. That is, we want an algorithm which offers the *computational speed* of linear techniques and the *interpretative advantage* of tree-structured rules. This immediately suggests an obvious solution—recursively partition the variable-space using linear discriminant functions. If successful, this would be an alternative to CART when linear combinations splits are used. Further, if observation (2) in the preceding subsection bears out, the solution may be reasonably accurate too.

Several difficulties have to be overcome before this idea is viable. We give two of them here. First, recursively partitioning the space (and the learning sample) using discriminant functions will quickly lead to almost singular covariance matrices in the subsamples (because the subsamples tend to be more homogeneous and hence reside in lower dimensional subspaces). Second, some data sets are just not amenable to linear partitions. An example is two spherically distributed classes, with one class entirely surrounding the other (see Section 3.4).

FACT overcomes the first difficulty by applying LDA only to the larger principal components at each node. To surmount the second, FACT tests for spherical structure at each node; it automatically looks for splits based on polar coordinates whenever the test is positive.

After a procedure for selecting splits is found, there is still the question of when to stop the splitting. The idea of pruning in CART is very clever, but it requires more computations than would a direct stopping rule. FACT simply uses a set of intuitive rules to stop splitting. The

examples in this paper will show that this works quite well.

To construct a tree which lends itself to easy interpretation, CART uses *univariate splits*. FACT offers the same option, with the splits now chosen via the standard *F-ratios* of "between groups" versus "within groups" variance.

As a result, FACT enjoys *both* the computational efficiency of linear methods *and* the tree-structured interpretative capability of CART. Because the FACT tree can have up to  $J$  splits at each node, the chance of some classes being unrepresented is reduced. Finally, because an explicit stopping rule is used, the FACT tree is nonrandomized.

FACT has been implemented in a FORTRAN computer program. It uses a number of subroutines from EISPACK (Smith et al. 1976), as well as some of the computing algorithms recommended by Chan et al. (1983). The program has been tested on many real and simulated data sets. Compared to CART, FACT runs many times faster, appears comparable in classification accuracy, and offers all the features present in CART (e.g. CV estimate of error; unequal priors; unequal misclassification costs; variable importance ranking; and class probability trees), as well as some features not available in CART (e.g. polar coordinate splits; splits with variables of mixed type; and the option of reducing run time ten times by giving up the CV estimate of error).

The rest of this paper describes the details and discusses the performance of FACT.

## 2. SALIENT FEATURES OF THE ALGORITHM

### 2.1. Priors and Misclassification Costs

CART allows the user to either estimate the class priors  $\{\pi(j), j = 1, \dots, J\}$  from the learning sample or have them specified in advance. The estimated posterior class probabilities  $\{p(j|t), j = 1, \dots, J\}$  at each node  $t$  are then given by

$$p(j|t) = p(j, t) / \sum_i p(i, t),$$

where  $p(j, t) = \pi(j)N_j(t)/N_j$ ,  $N_j$  is the number of class  $j$  objects in the learning sample, and

$N_j(t)$  is the number of class  $j$  objects in node  $t$ . Let  $C(i|j)$  be the cost of misclassifying as class  $i$  a class  $j$  object, and assume that  $C(i|j) = 0$  if  $i = j$ , and nonnegative otherwise. CART allows the user two ways to deal with unequal misclassification costs. One option is called "symmetric Gini", which uses the measure of node impurity  $\sum_{j,i} C(i|j)p(i|t)p(j|t)$  (Breiman et al. Section 4.4). The effect of this is to symmetrize the given cost matrix. The other option is called "altered priors", which converts the given unequal misclassification costs into unit costs by altering the priors (estimated or otherwise) to

$$\pi'(j) = C(j)\pi(j) / \sum_i C(i)\pi(i), \quad (2.1)$$

where  $C(j) = \sum_i C(i|j)$ . This conversion is exact if, for each class  $j$ , the misclassification cost  $C(i|j)$  is constant for every  $i \neq j$ ; otherwise it is approximate.

FACT allows the priors to be pre-specified or estimated too. And there are also two options for dealing with unequal misclassification costs. The first is through altered priors, the same as in CART. The second is described in the next subsection. Because it is similar in spirit to classical discriminant analysis assuming normality, it will be called "normal theory" in the sequel. With unequal misclassification costs, this option does not lead to linear splits, and so does not yield as simple a tree structure as with altered priors. However, it is less artificial in the face of nonconstant  $C(i|j)$  (over  $i \neq j$  for each  $j$ ).

The following formulas apply to the altered priors option as well, except that then  $C(i|j)$  would be taken as 1 for  $i \neq j$ , and the priors changed according to (2.1).

## 2.2. Splitting Rule

CART allows the user to choose either univariate or linear combination splits. In addition to these two options, FACT offers a third: linear-cum-polar coordinate splits. Because linear combination splits are more flexible than univariate splits, trees grown using the former usually have better classification accuracy than trees which use the latter. On the other hand, trees with univariate splits are easier to interpret.

CART uses a node impurity function to evaluate splits, the best split being found via exhaustive searches. To avoid the latter, FACT uses discriminant functions to split each node. Thus, when linear combination splits are desired, FACT essentially performs *piecewise* LDA. To overcome the problem of near-singular covariance matrices mentioned earlier, a principal components analysis (of the correlation matrix) is first made at each node. Linear discriminant functions are then calculated from those principal components whose eigenvalues exceed .05 times the largest eigenvalue. Besides avoiding singularity difficulties, this transformation step also permits some reduction of dimensionality.

Therefore when linear combination splits are desired, FACT selects its splits for each node  $t$  via linear discriminant functions which take the form

$$d_j(\mathbf{x}) = \hat{\mu}_j' \hat{\Sigma}^{-1} \mathbf{x} - \frac{1}{2} \hat{\mu}_j' \hat{\Sigma}^{-1} \hat{\mu}_j + \ln [p(j|t)], \quad (2.2)$$

where  $\hat{\mu}_j$  denotes the sample mean vector of the  $j$ th class and  $\hat{\Sigma}$  the pooled estimate of the covariance matrix (see e.g. Johnson and Wichern 1982, p. 499). Each node is split into  $J$  subnodes, and an object is channeled into the  $i$ th subnode if the latter minimizes the estimated expected misclassification cost:

$$\sum_{j=1}^J C(i|j) \exp[d_j(\mathbf{x})] = \min_{1 \leq m \leq J} \sum_{j=1}^J C(m|j) \exp[d_j(\mathbf{x})].$$

(This is the optimal strategy if we have normal densities in each node, cf. Anderson 1984.) In the case of equal misclassification costs, this rule reduces to: Put  $\mathbf{x}$  into the  $i$ th subnode if  $d_i(\mathbf{x}) = \max_j d_j(\mathbf{x})$ .

When univariate splits are desired, FACT selects the variable (i.e. coordinate) with the largest F-ratio of between-groups versus within-groups variance. If this F-ratio is  $\geq 4$ , FACT uses the same discriminant functions as above to split the node, except that the quantities in (2.2) refer to values along this coordinate only. (The idea behind this is similar to that in stepwise discriminant analysis as implemented in standard statistical packages such as BMDP, SAS or SPSS: variables are entered only if their F-ratios exceed a specified threshold value. The value we use is 4, the

same as in BMDP). Otherwise, each variable  $x_i$  is transformed into  $z_i = |x_i - \bar{x}_i|$ , the absolute deviation of  $x_i$  about the node mean for the  $i$ th coordinate. The F-ratios based on the  $z_i$ 's are then computed. Let  $z_{i^*}$  be the one with the largest F-ratio,  $F_{i^*}$ , say. If  $F_{i^*} \geq 4$ , the node is split using (2.2) with  $\mathbf{x} = z_{i^*}$ , etc. Otherwise, the node is simply split into two according as  $x_{i^*} \leq$  or  $> \bar{x}_{i^*}$ —the idea here is to break the symmetry in this node with the hope of getting good splits later.

When polar coordinate splits are allowed (in addition to linear combination splits), the following sequence of steps is taken by FACT at each node. As in the preceding paragraph, a check is first made to see if the largest F-ratio of the  $x_i$ 's is  $\geq 4$ . If affirmative, linear combination splits are used at that node. Otherwise, FACT transforms each  $x_i$  into a  $z_i$  as defined above. Let  $z_{i^*}$  be the  $z_i$  with the largest F-ratio. If this ratio is  $\geq 4$ , the node is split on  $z_{i^*}$ . If the ratio is less than 4, Levene's (1960) test of homogeneity of variances of the  $x_i$  is performed. Suppose  $m$  variables are found significant. If  $m = 1$  and  $x_{i^*}$  is the significant variable, the node is split according to  $x_{i^*} \leq$  or  $> \bar{x}_{i^*}$ . If  $m \neq 1$ , the  $x$ -variables are transformed to polar coordinates  $(r, \theta_1, \dots, \theta_{l-1})$  and the best univariate split from the  $z$ 's and  $r$  and  $\theta$ 's is found. Here  $l = m$  if  $m > 1$ , and  $l = k$  if  $m = 0$ . Again F-ratio-like quantities are used to determine the goodness of a split, except that we use the formulas in Watson (1983, pp. 5-6) for the "average" and "dispersion" of a set of angles.

Whether univariate, linear, or linear-cum-polar splits are used it is clear that:

- (1) FACT potentially generates  $J$  splits at each node (compared to two for CART). As a result, if  $J$  is much larger than 2, the typical FACT tree is broader, while the CART tree is longer.
- (2) FACT is quicker in choosing its splits than CART. The extra effort needed for FACT to find principal components, in the case of linear combination splits, is more than equaled by the much greater effort required to search exhaustively for these splits in CART.

We have assumed in the preceding discussion that the original variables are all ordered. When categorical variables are present, FACT first transforms each of them into an ordered variable. If  $x_i$ , say, is a categorical variable with  $c$  categories, FACT converts it to  $c-1$  dummy variables (i.e. 0-1 variables). The largest CRIMCOORD (or discriminant coordinate, see Gnanadesikan 1977, p. 86),  $y_i$  say, from this  $(c-1)$ -dimensional space is obtained. Each  $(c-1)$ -dummy vector is then transformed into this 1-dimensional  $y_i$ . Finally  $x_i$  is replaced by  $y_i$  in the learning sample. The 0-1 nature of the dummy variables makes it easy to re-express a split in terms of  $y_i$  to a question of the form "Is  $x_i$  in  $A$ ?", where  $A$  is a subset of the set of all categories. (The reason for using only the first CRIMCOORD is to ensure that a 1-dimensional categorical variable is mapped into a 1-dimensional ordered variable, thus preventing the dimension of the transformed space from increasing.)

In CART, categorical variables are handled differently from ordered ones. The algorithm searches through all possible subsets  $A$  to find the best split based on  $x_i$  for each  $i$ . Because CART does not convert a categorical to an ordered variable, the two types cannot appear together in the same linear combination split, i.e. the splits are either linear combinations of only ordered variables, or univariate splits on categorical variables. FACT, however, can combine all types of variables in a linear combination split.

### 2.3. Stopping Rule

CART does not use a stop-splitting rule. Instead, it grows a very large tree and then uses CV to prune the tree back to its eventual size. FACT employs the stopping rule: "*Stop splitting if either the node apparent error rate does not decrease with splitting, or there is at most one class in the node for which the number of samples is  $\geq$  MINDAT, a user-specified constant*". (MINDAT is either 5 or 10 in the examples in this paper.) It is well known that the apparent error rate (or *resubstitution estimate* in the terminology of Breiman et al.), is not a good estimate for the true error rate. It was considered in Breiman et al. (pp. 93-98) as a possible node impurity

function for selecting splits, but was found to be bad. FACT, too, does not use it to select splits. Because the apparent error rate is not used to select the splits, it does not necessarily decrease monotonically with the size of the tree. On the contrary, it has proven to be quite satisfactory as a stopping rule. (See Efron 1983, Table 1, for some empirical evidence that the apparent error rate is not always downward-biased for LDA.)

Specifically, a node  $t$  will be declared terminal if the resubstitution estimate of the expected misclassification cost is not decreased with splitting. Let  $t_1, \dots, t_J$  be the daughter nodes of  $t$  if it is split, and let  $l(t)$  denote the class that will be assigned to node  $t$  if it is declared terminal. Then splitting will stop at  $t$ , i.e.  $t$  will be judged terminal, if

$$\sum_{i=1}^J C(l(t)|i) p(i|t) \leq \sum_{j=1}^J [\sum_{i=1}^J C(l(t_j)|i) p(i|t_j)].$$

This rule shows that FACT is one-step optimal and nonrandomized.

#### 2.4. Terminal Node Assignment Rule

This rule is the same as in CART: Assign terminal node  $t$  to class  $i$  if

$$\sum_{j=1}^J C(i|j) p(j|t) = \min_{1 \leq m \leq J} \sum_{j=1}^J C(m|j) p(j|t),$$

i.e. a node is assigned to the class that minimizes the estimated expected misclassification cost.

This reduces to the plurality rule when misclassification costs are equal.

### 3. SOME EXAMPLES

This section demonstrates the performance of CART and FACT on five simulated examples. Equal priors are used throughout and, except in Section 3.6, unit misclassification costs are assumed. All the data in both learning and test samples are complete (i.e. without missing observations). Unless stated otherwise, the Gini criterion is used for choosing the splits in CART. Like CART, FACT uses ten-fold CV to obtain its CV estimate of error. The timings on FACT reported here include the times for CV. (The times are the CPU times on a Pyramid 90X super-

minicomputer with a floating point accelerator and running the 4.2 BSD UNIX operating system.)

The first example concerns normally distributed variables. The second and third examples are from Breiman et al., and the fourth is from Friedman (1977). The fifth example involves categorical variables with more than two categories each. The following abbreviations apply to all the tables in this paper:

CV Est. = "CV estimate of error"

TS Est. = "test sample estimate of error"

SE/Speed = "estimated standard error" or "relative speed of FACT to CART"

The results below are obtained with a MINDAT value (see Section 2.3) of 10 for example 2, and 5 for the other examples.

### 3.1. Normal Discrimination Problem

There are ten variables and three classes. Each class contains objects whose  $\mathbf{x}$ -vectors are normally distributed with the same identity covariance matrix, but with means situated at the corners of an equilateral triangle with sides of length 3 in the space spanned by the first two variables. Thus only the first two variables contain information, the other eight variables being pure noise. The asymptotic Bayes error rate is

$$\{1 - \Phi(3/2)\} + \int_{3/2}^{\infty} \phi(x) \Phi\{3^{-1/2}(3-x)\} dx = .1153.$$

where  $\phi(x)$  and  $\Phi(x)$  are the standard normal density and distribution functions respectively.

Table 1 shows how CART and FACT perform on this problem for  $n = 300$ . The first split in FACT is exactly as in LDA. Stepwise LDA via BMDP7M (Dixon et al. 1983) took 22s to execute, and the resulting classifier has a test sample estimate of error of  $.12 \pm .01$ . Hence FACT is

Table 1. Three-Class Normal Problem;  $k = 10$ ;  $J = 3$ ;  $n = 300$ ; Test sample size = 3000.

Criteria	CART	FACT	SE/Speed	CART	FACT	SE/Speed
	(1) Univariate Splits			(2) Linear Combination Splits		
CV Est.	.15	.13	$\pm .02$	.11	.11	$\pm .02$
TS Est.	.15	.16	$\pm .01$	.15	.14	$\pm .01$
Run time	210s	12.7s	16.5	2040s	69.7s	29.3

only slightly slower and less accurate than BMDP in this example. The error rates for CART and FACT are not significantly different in this example, but FACT runs about 16 times faster than CART with univariate splits and 30 times faster with linear combination splits.

### 3.2. Waveform Recognition Problem

This example is described in detail in Breiman et al. (pp. 49-55). There are 3 classes and 21 variables. Each class consists of a random convex combination of two triangular waveforms, with Gaussian noise added. Breiman et al. report that a test sample estimate of the error rate of the Bayes rule for this problem is .14, and observe that, with univariate splits, CART gives an error rate about twice the asymptotic Bayes rate.

Table 2 presents the results from a learning sample of size 300. In this case FACT is significantly better than CART on both accuracy and speed with either univariate or linear combination splits. (Breiman et al., p. 134, report a test sample estimate of .20 for CART with linear combination splits. This is quite different from the .30 that we get here. When another learning sample was used, the estimate for CART changed to .24, and that for FACT to .21.) As with the previous example, it takes CART about three times as long to run with *univariate* splits as it takes FACT to run with *linear combination* splits.

### 3.3. Digit Recognition Problem

This is the other running example in Breiman et al. There are 7 indicator variables, each denoting whether a light is on or off in the 7 lines that make up a digital display on an electronic watch. (See the top left corner of Figure 1.) Thus  $J = 10$ . Each light has probability .1 of not

Table 2. Waveform Recognition Problem;  $k = 21$ ;  $J = 3$ ;  $n = 300$ ; Test sample size = 2000.

Criteria	CART	FACT	SE/Speed	CART	FACT	SE/Speed
	(1) Univariate Splits			(2) Linear Combination Splits		
CV Est.	.31	.31	$\pm .03$	.24	.21	$\pm .02$
TS Est.	.31	.27	$\pm .01$	.30	.20	$\pm .01$
Run time	456.4s	25.2s	18.1	55.2m	2.4m	23

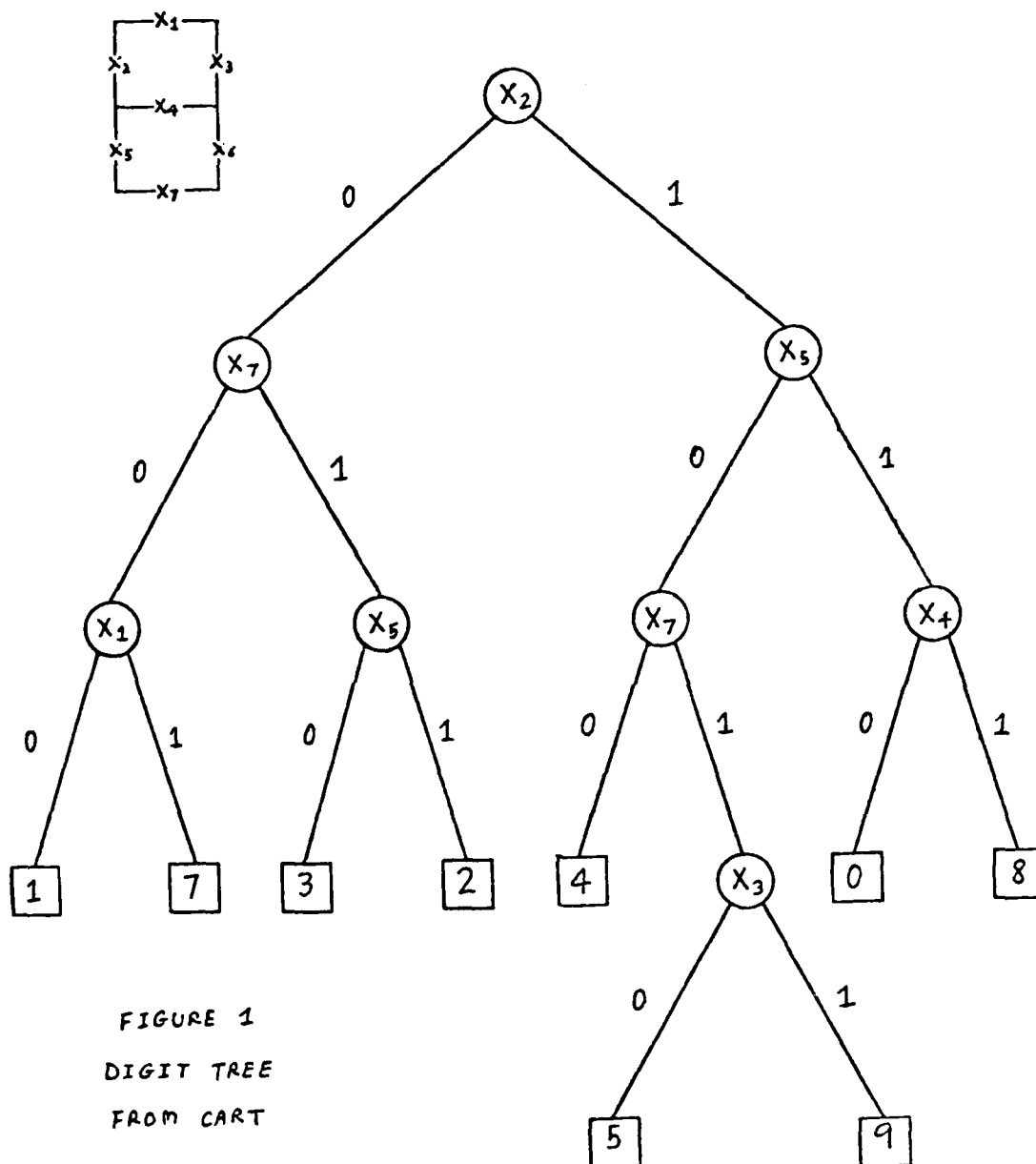


FIGURE 1  
DIGIT TREE  
FROM CART

Table 3. Digit Recognition Problem;  
 $k = 7$ ;  $J = 10$ ;  $n = 200$ ; Test sample size = 5000.

Criteria	CART	FACT	SE/Speed
(1) Univariate Splits			
CV Est.	.31	.30	$\pm .03$
TS Est.	.36	.32	$\pm .01$
Run time	77.2s	4.9s	15.8

doing what it is supposed to do, independently of the other lights. The asymptotic Bayes error rate is .26 (see Breiman et al., p. 47).

Although the variables may legitimately be taken to be ordered, they are assumed to be categorical in the results reported here. Figure 1 shows the CART tree constructed from a learning sample taken from a demonstration package which is distributed with the commercial CART software. The twoing criterion is used to grow the CART tree. The figure shows that no terminal node is assigned to the class for digit number "6". Figure 2 shows the corresponding FACT tree. Note that although the FACT algorithm splits every node into 10 subnodes, the 0-1 nature of the variables forces all the samples down only two subnodes at a time. If no samples appear in a subnode, FACT eliminates it in the final tree output. Thus FACT also produces a binary tree in this example.

Table 3 shows the accuracy and speed of FACT. The slightly higher test sample estimate of error for CART is probably because one class is not represented in the tree. When other learning samples are tried, the differences in accuracy between CART and FACT are not so significant (see Table 10 for the results with another learning sample). Linear combination splits are not tried because CART disallows this option when all the variables are categorical.

### 3.4. Spherical Distribution Problem

This example appears in Friedman (1977) and represents a case where LDA is ineffective. There are two classes and ten variables. The first four variables of one class are distributed uniformly within a four-dimensional spherical slab centered at the origin with inner radius 3.5 and

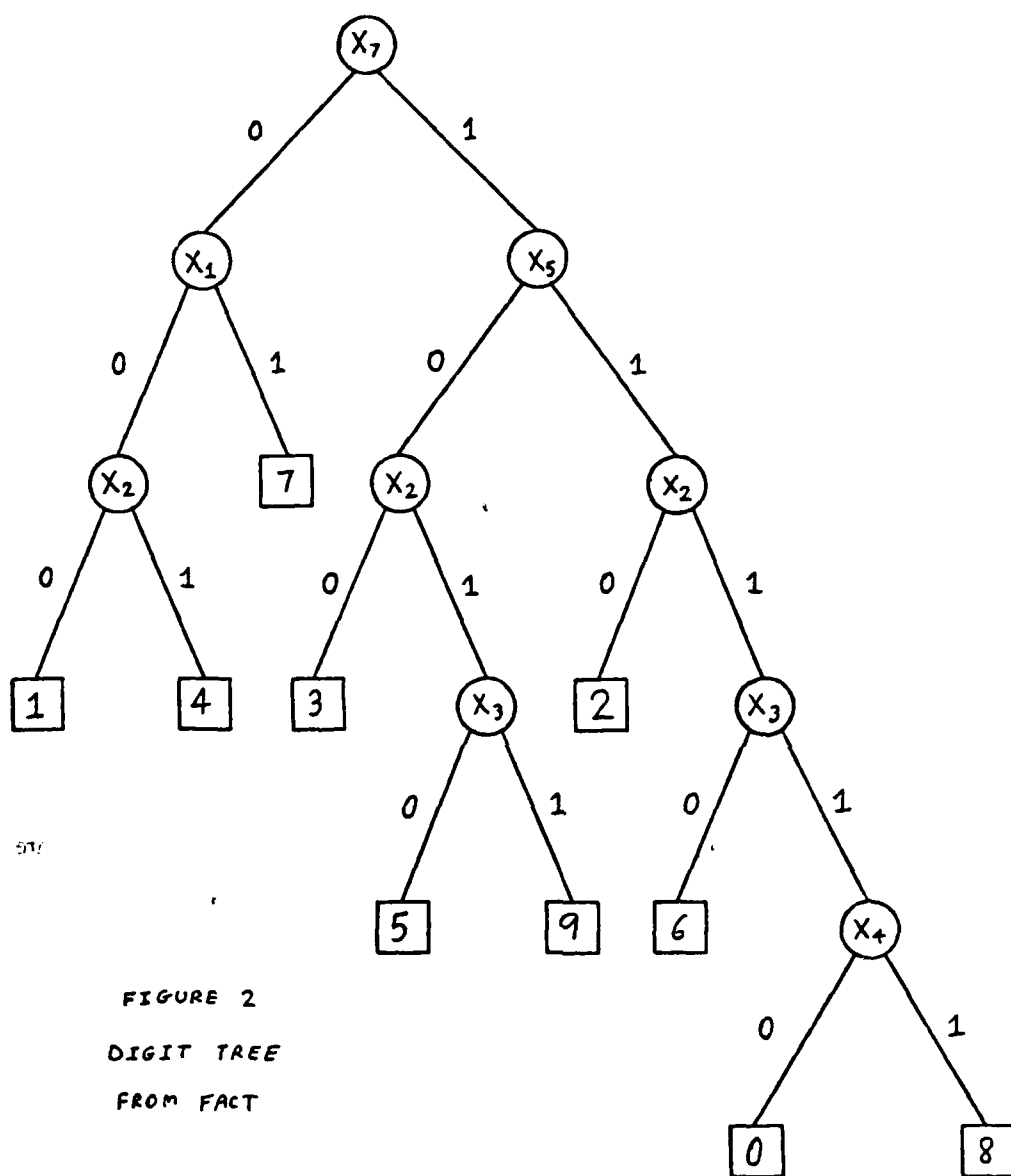


FIGURE 2  
DIGIT TREE  
FROM FACT

Table 4. Spherical Distribution Problem;  $k = 10$ ;  $J = 2$ ;  $n = 1000$ ; Test sample size = 5000.

Criteria	CART	FACT	SE/Speed	CART	FACT	SE/Speed
	(1) Univariate Splits			(2) Linear & Polar Splits		
CV Est.	.08	.18	$\pm .01$	.12	.06	$\pm .01$
TS Est.	.09	.19	$\pm .005$	.12	.05	$\pm .004$
Run time	898s	69.5s	12.9	2h	87s	80.

outer radius 4.0; the last six variables are independent and identically distributed standard normal. The variables in the other class are distributed as a ten-dimensional multivariate normal centered at the origin with identity covariance matrix. Thus, in the space containing the first four variables, the first class almost completely surrounds the second. Following Friedman (1977), we remove the spherical symmetry by scaling each coordinate by its sequence number, i.e. the first coordinate is divided by one, the second coordinate by two, and so on. The asymptotic Bayes error rate for this problem is .0063.

Table 4 reports the results from a learning sample of size 1000. CART is more accurate than FACT if univariate splits are used, although it takes about thirteen times as long as FACT to obtain the tree. (The better accuracy of CART here may be attributed to its use of CV to prune.) When linear-cum-polar splits are allowed, however, FACT is able to detect the spherical structure in the problem and the tree has only one split—along the radius. FACT takes just over one minute with this option, while CART runs for about two hours with linear combination splits.

It is interesting to note that CART is worse with linear combination splits than it is with univariate splits. This may be the combined result of pruning and the one-step optimality of the procedure, i.e. the tree with linear combination splits may be overpruned, and the best univariate split at a node may yield better splits later on than the best linear combination split.

### 3.5. Categorical Variable Problem

This problem has 3 classes and 5 variables all of which are strictly categorical. Variables  $x_1, \dots, x_5$  have 8, 3, 3, 3, and 10 categories respectively. Each variable takes values 1, 2, etc.

Table 5. Categorical Variable Problem;  
 $k = 5; J = 3; n = 300$ ; Test sample size = 5000.

Criteria	CART	FACT	SE/Speed
(1) Univariate Splits			
CV Est.	.59	.61	$\pm .03$
TS Est.	.57	.60	$\pm .01$
Run time	814s	37.5s	21.7

up to the number of its categories. The class distributions are:

- Class 1:  $P(x_1 = i, x_2 = j, x_3 = k) = 1/72$   
 Class 2:  $P(x_1 = i, x_2 = j, x_3 = k) \propto (i + j + k)$   
 Class 3:  $P(x_1 = i, x_2 = j, x_3 = k) \propto (8 - i + j + k)$ .

Variables  $x_4$  and  $x_5$  are pure noise with the same (uniform) distribution over all three classes.

The asymptotic Bayes error rate is .5795.

Table 5 shows the results with univariate splits. CART and FACT have close to the asymptotic Bayes rate. CART is slightly more accurate, but FACT runs 22 times faster.

### 3.6. Unequal Misclassification Costs

We now present the results of some simulations on the waveform recognition problem with unequal misclassification costs. Following Breiman et al. (Section 4.5), two cost matrices are used. The first is

Table 6. Waveform Recognition Problem  
 $k = 21; J = 3; n = 300$ ; Test sample size = 2000; Cost matrix (3.1).

Criteria	CART		FACT		SE/Speed
	A. P.	S. G.	A. P.	N. T.	
(1) Univariate Splits					
CV Est.	.35	.34	.63	.63	± .05
TS Est.	.54	.56	.54	.54	± .03
Run time	552s	612s	26.5s	33.9s	16-23
(2) Linear Combination Splits					
CV Est.	.32	.32	.42	.42	± .06
TS Est.	.61	.61	.33	.33	± .03
Run time	65.6m	90.2m	5.1m	5.1m	13-18

Table 7. Waveform Recognition Problem  
 $k = 21; J = 3; n = 300$ ; Test sample size = 2000; Cost matrix (3.2).

Criteria	CART		FACT		SE/Speed
	A. P.	S. G.	A. P.	N. T.	
(1) Univariate Splits					
CV Est.	.61	.59	.79	.55	± .07
TS Est.	.79	.72	.69	.72	± .03
Run time	408s	600s	48.2s	46.6s	8-13
(2) Linear Combination Splits					
CV Est.	.53	.55	.55	.53	± .06
TS Est.	.61	.72	.47	.47	± .02
Run time	49.3m	91.9m	6.8m	6.1m	7-15

$$C(i|j) = \begin{bmatrix} 0 & 5 & 5 \\ 1 & 0 & 1 \\ 1 & 1 & 0 \end{bmatrix}, \quad (3.1)$$

where  $i$  is the column and  $j$  the row index.

Table 6 gives the estimated accuracies and run times for both CART and FACT using the various options. (The abbreviations "A. P.", "S. G.", and "N. T." stand for altered priors, symmetric Gini, and normal theory, respectively.) Because the misclassification costs are constant for each row in (3.1), FACT gives the same trees under either option (although the computation times are different). FACT is between 16 to 23 times faster than CART with univariate splits. The corresponding figures with linear combination splits are 13 to 18. In terms of accuracy, FACT is similar to CART with univariate splits, but its misclassification cost is about half that of CART when linear combination splits are used.

Table 7 gives the results with the cost matrix:

$$C(i|j) = \begin{bmatrix} 0 & 3 & 3 \\ 3 & 0 & 1 \\ 3 & 1 & 0 \end{bmatrix}. \quad (3.2)$$

Because this matrix is symmetric, the symmetric Gini option may be expected to do better than altered priors for CART. The table indicates, however, that (for the present example at least) this is true only with univariate splits. Furthermore, although the altered priors option is less natural here, the results for FACT do not show it to be inferior to the N. T. option.

## 4. VARIABLE IMPORTANCE RANKING

### 4.1. The Problem of Masking

There is one danger in drawing conclusions from any tree structure, especially if univariate splits are used, because the tree can look deceptively simple. For example, it is tempting to conclude that only those variables which appear on the tree are important. Real data sometimes exhibit proxy phenomena, i.e. where two or more variables essentially measure the same thing. When univariate splits are requested, however, only one of these variables can be selected in each split. The others will not appear, i.e. they are "masked". Thus, one cannot assume that the variables which actually appear on a tree are the only or even the most important ones.

In CART, the problem of masking (as well as the problem of missing observations) is solved via the use of "surrogate splits" at each node. Suppose at node  $t$ ,  $s(t)$  is the best split, and it is based on variable  $x_{i^*}$ . A surrogate split  $s_i(t)$  based on variable  $x_i$ , for  $i \neq i^*$ , is the univariate split on  $x_i$  which best predicts the split  $s(t)$ . The measure of importance for variable  $x_i$  is the total decrease in tree impurity if the surrogate splits  $s_i(t)$  are used at each intermediate node  $t$ . (See Breiman et al., Section 5.3, for the exact definitions.)

FACT does not make use of surrogate splits. Instead, the importance of a variable is simply the weighted sum of its F-ratios at the nodes. Specifically, if  $F_{i,t}$  denotes the ratio of the between-groups to the within-groups variance of variable  $x_i$  at node  $t$ , and  $p(t)$  is the estimated probability that an object will fall into node  $t$  (i.e. the proportion of learning samples in the node), then the importance of  $x_i$  is defined to be  $\text{Imp}(x_i) = \sum_t F_{i,t} p(t)$ , where the sum is over all nonterminal nodes in the tree. Thus, no additional computations (like searching for surrogate splits) are needed to obtain an importance ranking of all the variables.

Table 8. Importance Ranking: Digit Recognition Problem

CART		FACT	
Var.	Imp.	Var.	Imp.
x4	100	x3	100
x2	71	x5	82
x5	69	x7	80
x1	40	x4	76
x3	30	x1	54
x7	21	x2	49
x6	20	x6	40

#### 4.2. Examples

Like CART, FACT standardizes the highest ranked variable to have weight 100. Table 8 reports the importance ranking for the digit recognition problem. In this example, all seven variables are independent and hence should ideally be ranked equally. As the table shows, FACT does slightly better here since the lowest weight it gives is 40 versus 20 in CART. Note that in both the CART and FACT trees (Figures 1 and 2) the variable chosen at the first split is not the highest ranked.

Table 9 gives the importance ranking for the spherical distribution problem. Recall that only the first four variables are informative, the last six constituting noise. The table shows that FACT gives much lower weights to the noise variables than does CART. (The ranking for FACT

Table 9. Importance Ranking: Spherical Distribution Problem

Univariate Splits				Linear & Polar Splits			
CART		FACT		CART		FACT	
Var.	Imp.	Var.	Imp.	Var.	Imp.	Var.	Imp.
x4	100	x2	100	x2	100	x4	100
x2	91	x1	92	x3	98	x3	75
x3	85	x3	79	x4	91	x2	66
x1	61	x4	62	x1	89	x1	64
x8	12	x7	1.1	x9	13	x7	1.1
x9	11	x8	.6	x5	12	x10	.5
x5	9	x10	.4	x7	10	x8	.2
x10	8	x5	.3	x10	9	x6	.1
x7	7	x6	.3	x8	8	x5	.1
x6	5	x9	.1	x6	7	x9	.0

with linear-cum-polar splits shows a big drop in importance between the highest and next-highest ranked variables because the tree has only one split.)

## 5. MISSING OBSERVATIONS

As already mentioned, CART handles missing observations via surrogate splits. In FACT, missing values in the learning sample are replaced with class means estimated from the nonmissing values (cf. *BMDPAM* in Dixon et al. 1983). If a new object to be classified has missing values, these values are estimated with the coordinate means of that class in the learning sample which is closest to the object, in the space of the nonmissing coordinates. While less sophisticated, this method has the advantage of computational efficiency. Of course, like most general methods of dealing with missing data, it may not be appropriate if the data are not missing at random, and the results in this section will probably not generalize to such cases.

Table 10 shows the effect of missing values on the digit recognition problem for  $n = 200$ . (The learning sample is different from that in Table 3.) Both the learning and test samples have

Table 10. CV and TS Estimates of Error for Digit Recognition Problem;  
 $p$  % Missing in Both Learning and Test Samples;  $SE(CV) = .03$ ,  $SE(TS) = .01$ .

(1) Univariate Splits				
$p$	CART		FACT	
	CV	TS	CV	TS
0	.29	.30	.31	.32
5	.33	.34	.31	.35
10	.36	.36	.35	.39
25	.47	.46	.52	.51

Table 11. CV and TS Estimates of Error for Waveform Recognition Problem;  
 $p$  % Missing in Both Learning and Test Samples;  $SE(CV) = .03$ ,  $SE(TS) = .01$ .

$p$	(1) Univariate Splits				(2) Linear Combination Splits			
	CART		FACT		CART		FACT	
	CV	TS	CV	TS	CV	TS	CV	TS
0	.31	.31	.31	.27	.24	.30	.22	.20
5	.35	.33	.29	.31	.34	.31	.25	.20
10	.31	.35	.27	.35	.33	.34	.23	.20
25	.38	.37	.28	.42	.40	.40	.29	.22

$p$  % of their values randomly deleted, with  $p = 0, 5, 10, 25$ . The test sample estimates indicate that CART is handling missing values slightly better than FACT. Table 11 shows the results for the waveform recognition problem. For univariate splits, CART is again better than FACT, but the conclusion is different for linear combination splits—here the FACT error rate is surprisingly stable over the values of  $p$  considered. (The CV estimate of .28 for FACT with univariate splits and 25% missing seems to badly underestimate the true error rate. This result is unchanged when the experiment is replicated with another random seed, although the CV estimate for CART also drops to .28. CART stratifies the sample so that the CV samples have the same proportion of missing values as the whole sample. Stratification has not been implemented in FACT yet. This may explain why FACT does not do as well when  $p$  is large.)

Breiman et al. (Tables 5.3 and 5.4) give similar results for CART with univariate splits. They note that because the variables in the digit problem are not highly correlated, good surrogate splits are harder to find than in the waveform problem. FACT seems to have a similar difficulty with univariate splits here.

Two further experiments were run to examine the separate effects of missing values in the learning and test samples for the digit recognition problem. The first experiment consists of having  $p$  % of the data missing from the learning sample but not the test sample. The second experiment is the opposite: missing values occur in the test sample but not the learning sample. Univariate splits are used throughout. Table 12 gives the results. The pattern is quite clear. For

Table 12. Test Sample Estimates of Error for Digit Recognition Problem;  
Missing Values in Either Learning or Test Samples; SE = .01.

First Experiment $p$ % Missing in Learning Sample			Second Experiment $p$ % Missing in Test Sample		
$p$	CART	FACT	$p$	CART	FACT
0	.30	.32	0	.30	.32
5	.32	.32	5	.33	.35
10	.30	.32	10	.36	.39
25	.31	.31	25	.44	.50

both CART and FACT, missing values are more damaging if they occur in the test sample than in the learning sample (cf. Breiman et al. Table 5.5).

The tentative conclusion from these experiments is that CART seems better than FACT in handling missing values with univariate splits, but not with linear combination splits. (See Section 7.2 for further comments.)

## 6. TREE INTERPRETATION AND EFFECT OF TRANSFORMATIONS: THE BOSTON HOUSING DATA

The emphasis in the preceding sections has been on the speed and accuracy of FACT. We now focus on the differences in interpretation between the CART and FACT trees by applying the two methods to a real data set. The same data are also used to illustrate the advantages and disadvantages of the non-invariant nature of FACT with respect to monotone transformations.

The criteria CART uses to select its splits make it invariant of monotone transformations in the individual ordered variables when univariate splits are used. FACT does not possess this property. It is not clear whether invariance is always desirable. Certainly, invariance can protect the user against poor choice of scale for the variables. On the other hand, the possibility of getting different trees with different transformations allows the user a little more flexibility in working with the data (cf. ordinary regression versus regression based on ranks). In any case, CART is not invariant with linear combination splits, and neither is FACT.

To see the effects of transformations on FACT, we reanalyze the data on 1970 Boston housing values reported by Harrison and Rubinfeld (1978) and used extensively in Belsley, Kuh and Welsch (1980). There are 506 cases (census tracts) and 14 variables in this data. One variable is the median value of homes in thousands of dollars (MV), and the others are:

CRIM:	crime rate
ZN:	% land zoned for lots
INDUS:	% nonretail business
CHAS:	1 if on Charles River, 0 otherwise

NOX:	nitrogen oxide concentration
RM:	average number of rooms
AGE:	% built before 1940
DIS:	weighted distance to employment centers
RAD:	accessibility to radial highways
TAX:	tax rate
P/T:	pupil/teacher ratio
B:	$(Bk - .63)^2$ , where $Bk$ = proportion of blacks in population
LSTAT:	% lower-status population

Harrison and Rubinfeld (1978) regress  $\ln(MV)$  on the other 13 variables. To set this up as a classification problem, we categorize  $MV$  into three categories: "low" (class 1) if  $\ln(MV) \leq 9.84$ , "high" (class 3) if  $\ln(MV) > 10.075$ , and "medium" (class 2) otherwise.

Figures 3 and 4 show the CART and FACT trees respectively using univariate splits, and the left half of Table 13 presents their importance ranking. (The triples beside each node refer to the composition of the node. For example, at the root node there are 167 class 1, 173 class 2, and 166 class 3 samples.) There are some similarities between the CART and FACT trees. Both split on variable LSTAT first. Inspection shows that this variable actually splits the sample into two pieces for CART, but five pieces for FACT. The next variables split on are RM and NOX (CART) or RM and AGE (FACT). CART splits on RM if  $LSTAT \leq 14.4$ , and FACT splits on

Table 13. Variable Importance Ranking: Boston Housing Data

CART		FACT				BMDP7M	
Var.	Imp.	Untransformed		Transformed		Untransf.	Transf.
Var.	Imp.	Var.	Imp.	Var.	Imp.	Var.	Var.
LSTAT	100	LSTAT	100	$\ln(LSTAT)$	100	LSTAT	$\ln(LSTAT)$
RM	69	RM	44	$RM^2$	39	RM	$RM^2$
P/T	62	NOX	34	AGE	30	P/T	P/T
INDUS	61	AGE	32	$NOX^2$	29	NOX	B
AGE	58	TAX	31	INDUS	28	DIS	$NOX^2$
CRIM	55	INDUS	31	TAX	28	B	$\ln(DIS)$
DIS	54	P/T	24	P/T	21	AGE	ZN
NOX	53	RAD	20	$\ln(DIS)$	18	ZN	AGE
TAX	53	CRIM	18	CRIM	16		
RAD	36	B	17	B	15		
B	34	ZN	14	$\ln(RAD)$	14		
ZN	12	DIS	14	ZN	11		
CHAS	5	CHAS	3	CHAS	1		

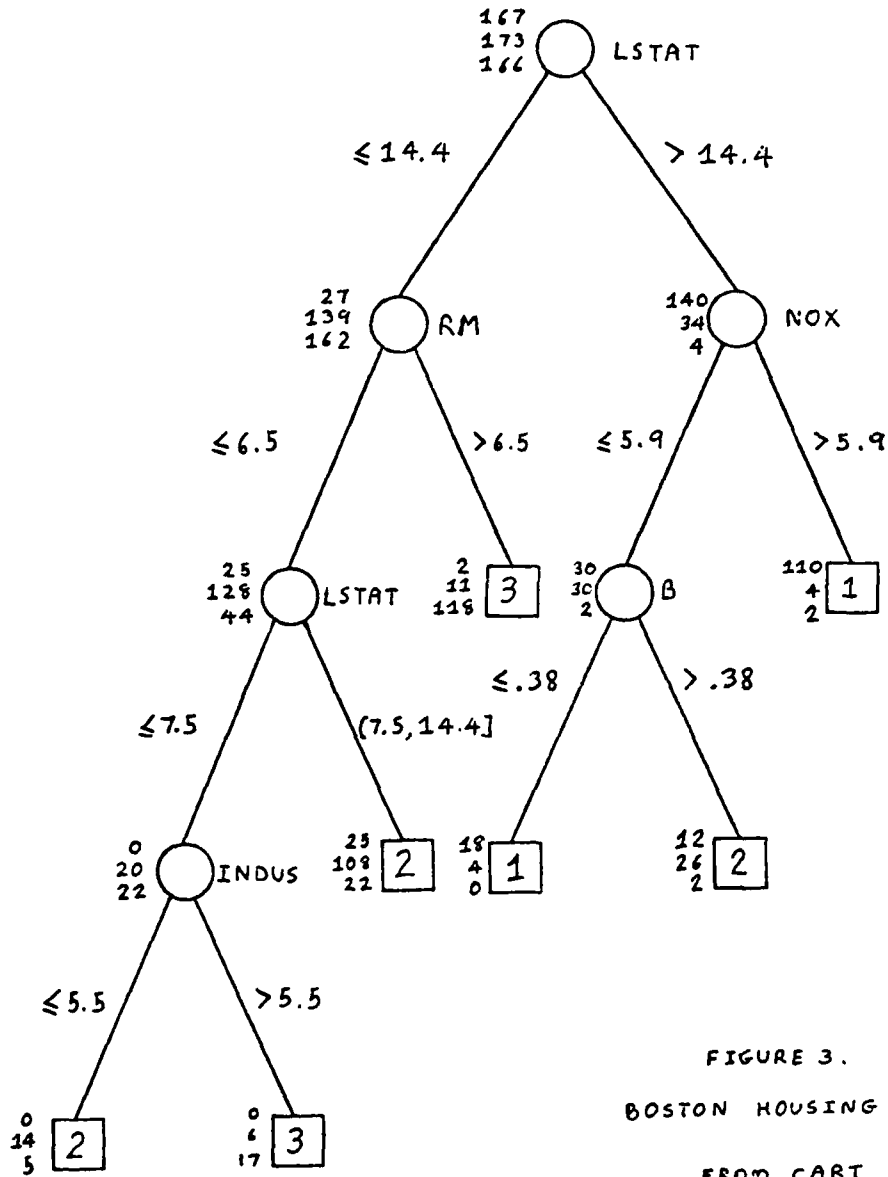


FIGURE 3.  
BOSTON HOUSING TREE  
FROM CART

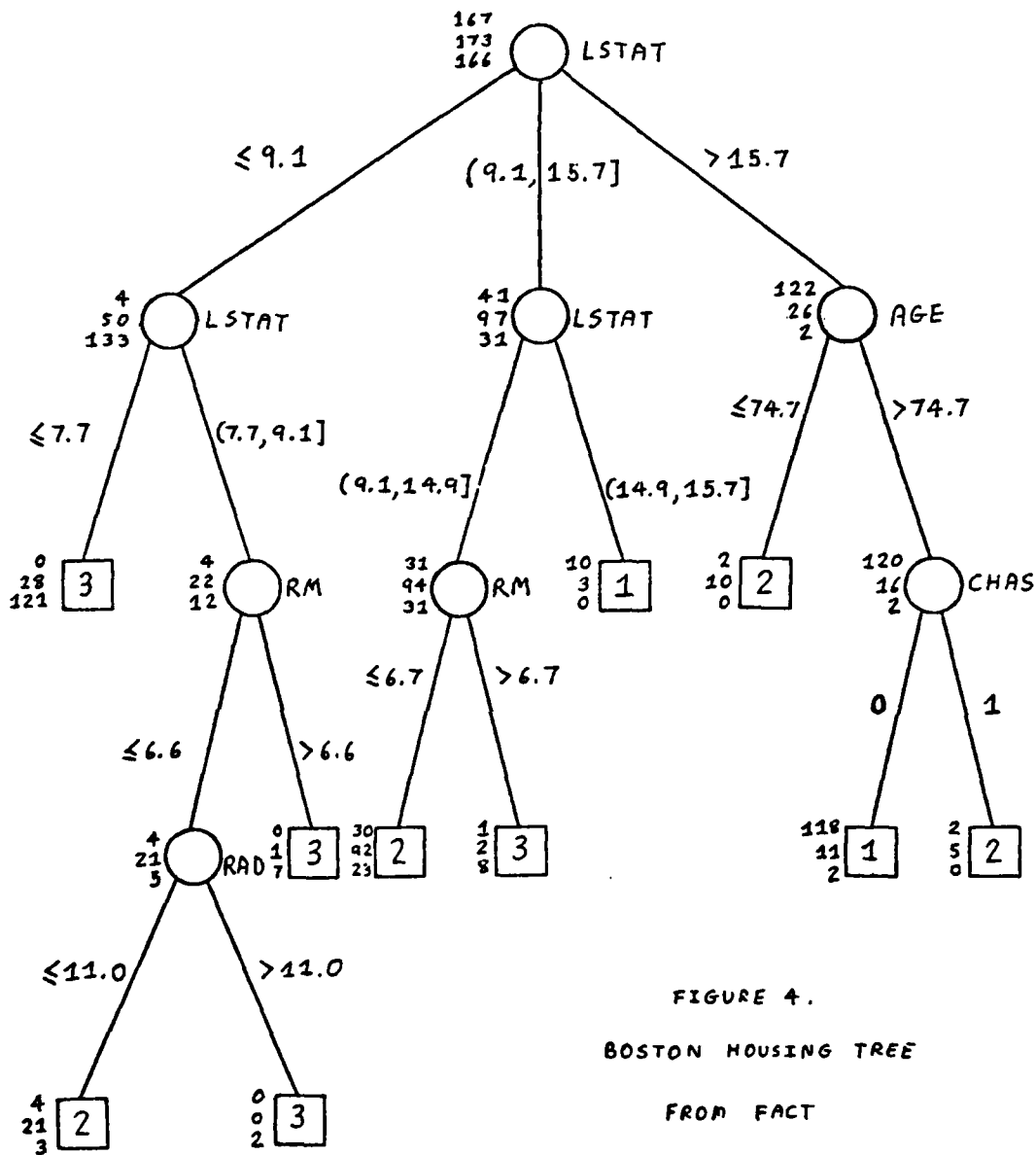


FIGURE 4.  
BOSTON HOUSING TREE  
FROM FACT  
(ORIGINAL VARIABLES)

RM if  $7.7 < \text{LSTAT} \leq 14.9$ . The trees indicate that NOX and AGE are proxies for each other if LSTAT is greater than about 15; this is also reflected in the importance rankings in Table 13 where both variables have similar importance measures. Variables CHAS, B and RAD are ranked relatively low by both CART and FACT, yet they appear in the trees. The node compositions where CHAS and RAD appear in the FACT tree show, however, that they generate noisy splits since each node is pre-terminal and almost all the sample goes into a single terminal node. These nodes may therefore be pruned back. It is not so obvious from the node composition in the CART tree that the split on B is noisy.

Five of the predictor variables are transformed in the regression model of Harrison and Rubinfeld, namely  $\ln(\text{LSTAT})$ ,  $\text{RM}^2$ ,  $\text{NOX}^2$ ,  $\ln(\text{RAD})$ , and  $\ln(\text{DIS})$  (see Belsley et al. p. 231 or Breiman et al. p. 218). Figure 5 shows how the FACT tree changes with these transformations (the CART tree is unchanged of course). The splits have been re-expressed in the original scale to facilitate comparison with Figures 3 and 4. Table 13 also gives the corresponding importance ranking. The FACT tree now has fewer terminal nodes (equal to CART). Further, one of the cut-points at the root node in Figure 5 is precisely the cut-point at the root node for CART. Examination of Figure 5 indicates that the FACT tree can essentially be pruned back to just 3 splits on LSTAT at the root node, since most of the sample is classified as class 3 if  $\text{LSTAT} \leq 8$ , class 2 if  $8 < \text{LSTAT} \leq 14.4$ , and class 1 otherwise. This is supported by Table 13, where the importance of LSTAT is much higher than that of the next highest ranked variable for all the methods.

We believe that tree interpretation is made easier when the number of splits per node is equal to the number of classes—especially if  $J$  is not small. When this does not occur at every node, understanding seems most assisted if this happens at the root node at least. (If the learning sample contains roughly equal numbers of objects per class, priors are estimated from the sample, and the variable chosen to split on at the root node is non-categorical, the FACT tree will have  $J$  splits at this node.) Figure 5 is a good illustration. There the learning sample is first split into 3

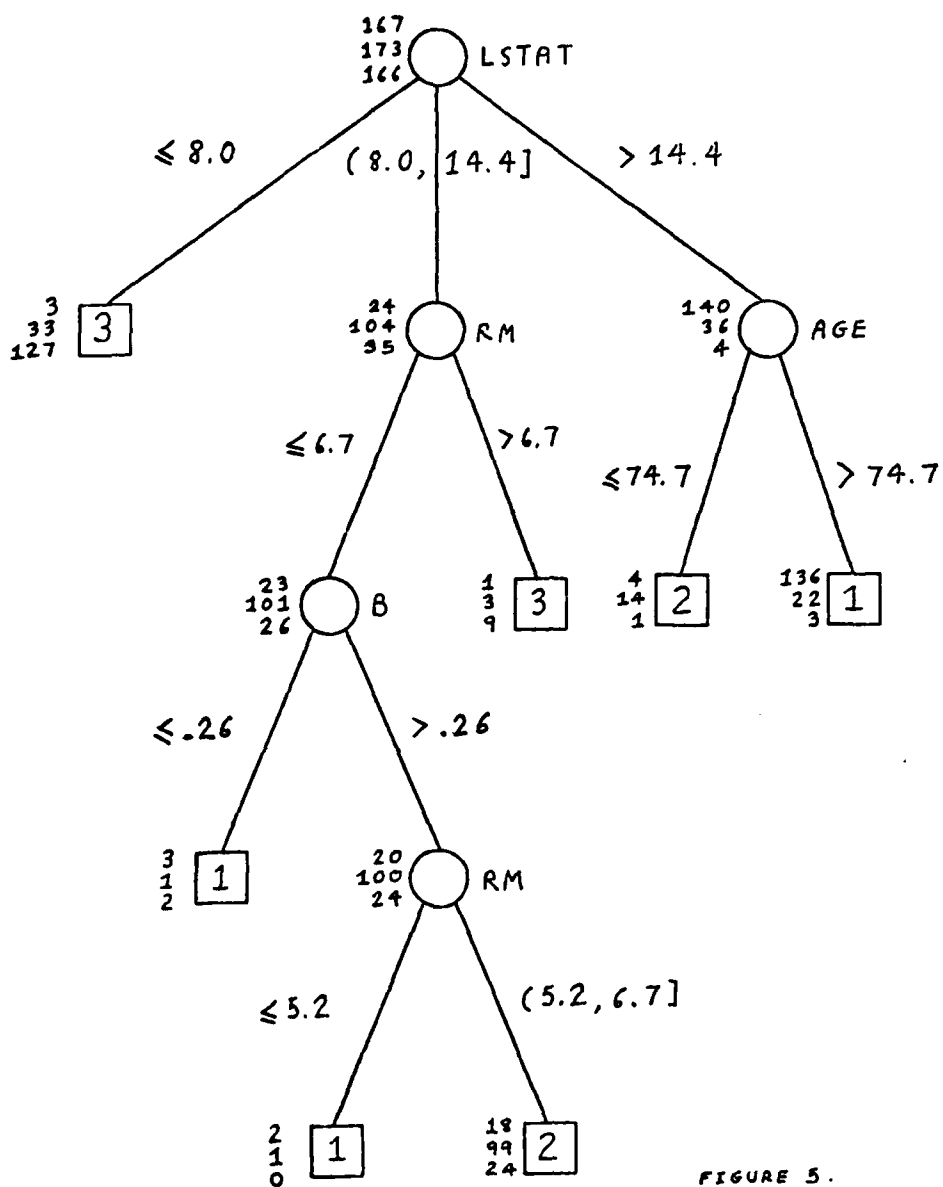


FIGURE 5.  
BOSTON HOUSING TREE  
FROM FACT  
(TRANSFORMED VARIABLES)

groups according to the value of LSTAT. This makes sense given the definition of this variable: (i) census tracts that are more affluent (left branch of tree) tend to have more expensive homes (class 3); (ii) for tracts which are much less affluent (right branch), the MV values are either low (class 1) or medium (class 2), depending on the age of the homes; (iii) finally, the node compositions along the middle branch indicate that tracts with average values of LSTAT are predominantly white ( $B > .26$  implies that  $Bk < .12$ ), and that for these tracts, housing values are mainly determined by the number of rooms. The reason that the nodes other than the root node do not have 3 splits each is because the discriminant functions (F-ratios) are weighted by the estimated class priors at the nodes. When these estimates are very disparate, as is the case here, splits can be produced where some of the daughter nodes contain no learning samples. Such empty nodes are combined with their neighbors, and the FACT tree will show less than  $J$  splits at these places.

For univariate splits, CART takes  $8\frac{1}{2}$  minutes to run, while FACT takes  $\frac{1}{2}$  minute for both original and transformed data. The CV estimates are  $.24 \pm .02$  (CART),  $.28 \pm .02$  (FACT, original scale), and  $.23 \pm .02$  (FACT, transformed scale), indicating that the transformations are useful. For linear combination splits (original as well as transformed data), CART takes  $1\frac{1}{2}$  hours and FACT takes 4 minutes. The corresponding CV estimates are  $.24$  (CART, original scale),  $.25$  (CART, transformed scale),  $.25$  (FACT, original),  $.27$  (FACT, transformed), with a common SE of  $\pm .02$ . For comparison, stepwise LDA via BMDP7M takes a little over  $\frac{1}{2}$  minute and gives a jackknife estimate of error of  $.25 \pm .02$  for original and  $.26 \pm .02$  for transformed data. Table 13 also gives the order in which BMDP7M enters the variables.

## 7. CONCLUDING REMARKS

The purpose of this paper is to find a fast alternative to CART which does not sacrifice classification accuracy. We briefly review the extent to which this is achieved.

### 7.1. Speed of Classifier Construction

The results in Tables 1 through 7 show that the speed of FACT is impressive. These tables yield a total of 24 ratios of run times of CART to FACT, with a minimum of 7 (Table 7), a maximum of 80 (Table 4), and a median of 16. As noted in point (4) of Section 1.3, these ratios may be multiplied by 10 if a CV estimate of error is not required—because ten-fold CV is used in all the examples, and FACT does not need it for tree construction.

### 7.2. Classification Accuracy

The test sample estimates of error in the tables show that neither CART nor FACT dominates on classification accuracy—CART wins 8 times, FACT wins 17 times, and both are tied 3 times. Often, however, the better accuracy of one method over the other is not statistically significant. Exceptions are Tables 2, 4, 6 and 11. These tables exhibit an apparent pattern: where CART is significantly better than FACT, this tends to be when univariate splits are used; the opposite seems to hold with linear combination splits, whether there are missing values or not. We conclude that (i) pruning is a powerful tool, and (ii) recursive use of linear discriminant functions can generate highly effective splits.

The above observations are admittedly based on a few examples, and the usual words of caution about generalizations to other situations are in order here. On the theoretical side, however, it is easy to see that the same limit theorem on the *Bayes risk consistency* of CART (proved in Breiman et al. 1984, Theorem 12.19), holds also for FACT. This theorem is very general, and is applicable to almost any sensible classification rule based on recursive partitioning.

### 7.3. Estimates of Error

FACT, like CART, gives a resubstitution estimate of error as well as a CV estimate. But because we believe the latter is more reliable, we do not consider the former here.

Figure 6 shows a plot of the CV estimates versus the test sample estimates of error reported in the above experiments. The CV estimates for CART tend underestimate the true error rate, while those for FACT appear to be more unbiased. (The two least-squares lines are also drawn on the plot.)

The apparent lack of unbiasedness of the CV estimate for CART may be because it is not a true cross-validation estimate of the error of the tree. As mentioned in point (4) of Section 1.3, cross-validation is used not only for error estimation but also for pruning the tree. That is, in addition to the main tree, a set of cross-validation trees are grown. The smallest cross-validation estimate from this set of trees is used to prune the main tree. This *same* estimate is then reported as the "CV" estimate in CART. A proper CV estimate would require another set of cross-validation trees to be grown and individually pruned just like the main tree. The computational requirements for this will be much greater than what it is now of course.

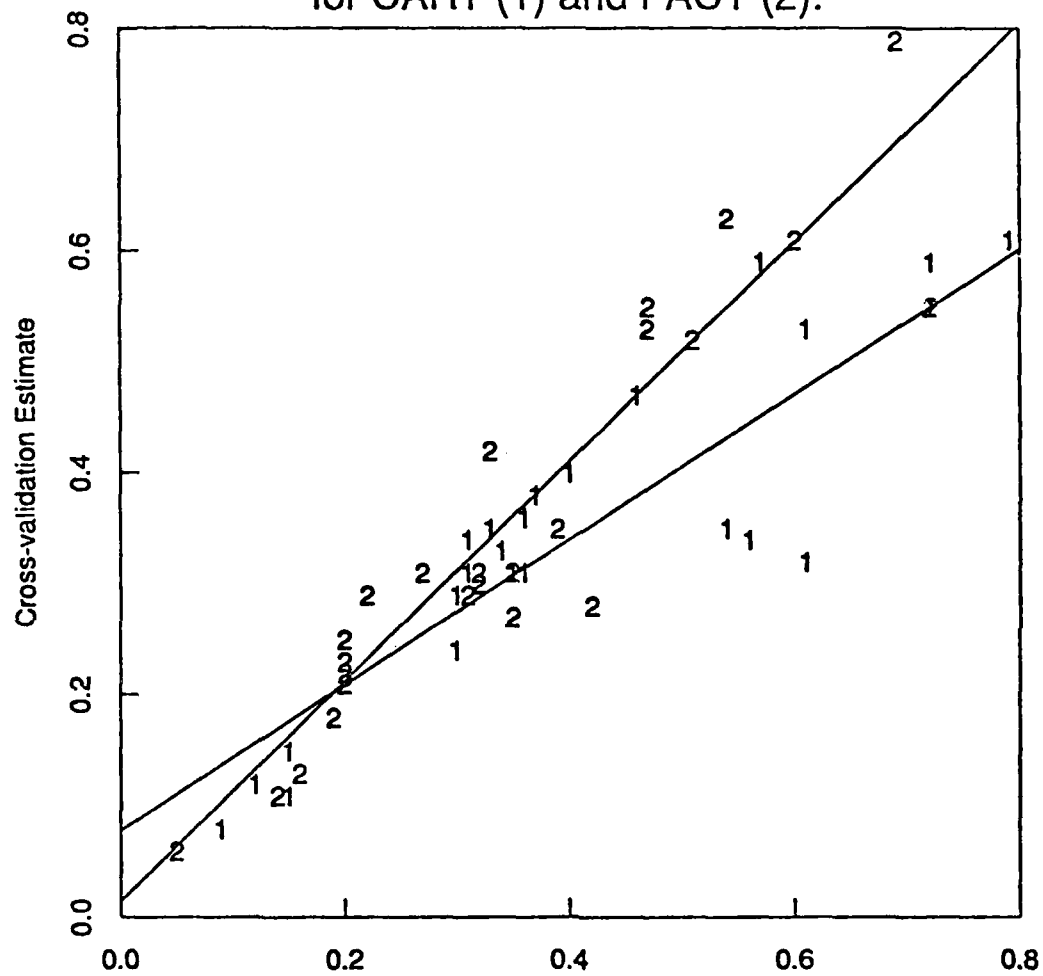
#### 7.4. Other Aspects

We summarize here other aspects in which FACT differs from CART:

- (1) FACT has the potential to give  $J$  splits per node, particularly at the root node (cf. (3) of Section 1.3).
- (2) FACT is nonrandomized (see (5) of Section 1.3).
- (3) FACT offers polar coordinate splits in addition to univariate and linear combination splits.
- (4) FACT also accepts missing observations and yields a variable importance ranking, but does so differently from CART.
- (5) FACT is not invariant of monotone transformations in the individual ordered variables.

Our overall conclusion is that FACT offers a reasonable alternative to CART, especially when speed is important or computational resources limited.

Figure 6. Plot of CV Versus TS Estimates  
for CART (1) and FACT (2).



The Two Least Squares Lines are Superimposed

## REFERENCES

- (1) Anderson, T. W. (1984), *An Introduction to Multivariate Statistical Analysis*, Second Edition, Wiley, New York.
- (2) Belsley, D. A., Kuh, E. and Welsch, R. E. (1980), *Regression Diagnostics*, Wiley, New York.
- (3) Breiman, L., Friedman, J. H., Olshen, R. A. and Stone, C. J. (1984), *Classification and Regression Trees*, Wadsworth, Belmont.
- (4) Chan, T. F., Golub, G. H., and LeVeque, R. J. (1983), "Algorithms for Computing the Sample Variance: Analysis and Recommendations," *American Statistician*, 37, 242-247.
- (5) Dixon, W. J., Brown, M. B., Engelman, L., Frane, J. W., Hill, M. A., Jennrich, R. I., and Toporek, J. D. (1983), *BMDP Statistical Software*, University of California Press, Berkeley.
- (6) Efron, B. (1983), "Estimating the Error Rate of a Prediction Rule: Improvement on Cross-validation," *Journal of the American Statistical Association*, 78, 316-331.
- (7) Fielding, A. (1977), "Binary Segmentation: The Automatic Interaction Detector and Related Techniques for Exploring Data Structure," in *The Analysis of Survey Data, Volume I, Exploring Data Structures*, (C. A. O'Muirheartaigh and C. Payne eds.), Wiley, New York.
- (8) Friedman, J. H. (1977), "A Recursive Partitioning Decision Rule for Nonparametric Classification," *IEEE Transactions on Computers*, C-26, 404-408.
- (9) Gnanadesikan, R. (1977), *Methods for Statistical Data Analysis of Multivariate Observations*, Wiley, New York.
- (10) Gordon, A. D. (1981), *Classification*, Chapman and Hall, London.
- (11) Hand, D. J. (1981), *Discrimination and Classification*, Wiley, New York.
- (12) Harrison, D. and Rubinfeld, D. L. (1978), "Hedonic Prices and the Demand for Clean Air," *Journal of Environmental Economics and Management*, 5, 81-102.
- (13) Johnson, R. A. and Wichern, D. W. (1982), *Applied Multivariate Analysis*, Prentice-Hall, Englewood Cliffs.
- (14) Levene, H. (1960), "Robust Tests for Equality of Variances," in *Contributions to Probability and Statistics*, (I. Olkin ed.), 278-292, Stanford University Press, Palo Alto.
- (15) Morgan, J. N. and Messenger, R. C. (1973), "THAID: A Sequential Analysis Program for the Analysis of Nominal Scale Dependent Variables," Institute for Social Research, University of Michigan, Ann Arbor.
- (16) Smith B. T., Boyle, J. M., Dongarra, J. J., Garbow, B. S., Ikebe, Y., Klema, V. C., and Moler, C. B. (1976), *Matrix Eigensystem Routines—EISPACK Guide*, Second Edition, Lecture Notes in Computer Science No. 6, Springer-Verlag, New York.
- (17) Watson, G. S. (1983), *Statistics on Spheres*, Wiley, New York.

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER #2902	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle)  TREE-STRUCTURED CLASSIFICATION VIA GENERALIZED DISCRIMINANT ANALYSIS		5. TYPE OF REPORT & PERIOD COVERED Summary Report - no specific reporting period
		6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s)  Wei-Yin Loh and N. Vanichsetakul		8. CONTRACT OR GRANT NUMBER(s) MCS 8300140 & DMS 8502303 DAAG29-80-C-0041
9. PERFORMING ORGANIZATION NAME AND ADDRESS Mathematics Research Center, University of 610 Walnut Street Madison, Wisconsin 53705		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS Work Unit No. 4 - Statistics and Probability
11. CONTROLLING OFFICE NAME AND ADDRESS  See Item 18 below		12. REPORT DATE January 1986
		13. NUMBER OF PAGES 33
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office)		15. SECURITY CLASS. (of this report)  UNCLASSIFIED
		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE
16. DISTRIBUTION STATEMENT (of this Report)  Approved for public release; distribution unlimited.		
17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)		
18. SUPPLEMENTARY NOTES U. S. Army Research Office P. O. Box 12211 Research Triangle Park North Carolina 27709  National Science Foundation Washington, DC 20550		
19. KEY WORDS (Continue on reverse side if necessary and identify by block number)  Classification trees; Cross-validation; Linear discriminant analysis; Misclassification costs; Missing values; Principal components; Recursive partitioning.		
20. ABSTRACT (Continue on reverse side if necessary and identify by block number)  Linear techniques are used recursively to construct classification rules which can be represented as <del>k</del> -ary decision trees. The method has been imple- mented in a computer program called FACT. It can handle ordered and unordered variables, unequal priors, variable misclassification costs, and missing observa- tions. Besides the tree structure, it also yields an importance ranking of the variables and a cross-validation estimate of error. FACT is compared with CART (a procedure proposed recently by Breiman et al., which gives a binary tree) in a series of examples. The conclusion is that FACT and CART are usually comparable in terms of classification accuracy and interpretative capability, but FACT runs many times faster.		

END

FILMED

6-86

DTIC